



# DIRAC Pilot 3.0

Andrew McNab

LHCb, GridPP,  
University of Manchester



# Overview

- Pilot 2.0 + modularity
- Pilot 3.0 + universality
- Bootstrapping pilot scripts
- Bootstrapping configuration
- Logging
- Pilot 3.0 repository
- VMs/SudoCE
- Singularity
- Other containers



## Pilot 2.0

- Pilot 2.0 introduced modular framework with “commands” (= functions) which pilots execute
- Python files provide libraries of commands, which can be selected in configuration
- This can be extended with extra Python files provide additional commands needed by a particular XyzDIRAC instance
  - Several XyzDIRAC instances (eg LHCb, GridPP, Belle II)
- `dirac-pilot.py` and files with commands are sent by the SiteDirector inside the input sandbox
- Various mechanisms have been used to bootstrap this for VMs
  - In particular, an SVN checkout



## Pilot 3.0

- Pilot 3.0 continues to extend the framework, with an emphasis on “universality”
- Converge SiteDirector and Vacuum VM scripts as much as possible
- Avoid/minimise the need for forking by different XyzDIRAC instances
- Provide a single logging framework for SiteDirector and Vacuum VM cases
- Decouple pilot version from main DIRAC version
  - Using the dedicated DIRACgrid/Pilot repository in GitHub

# “wget” bootstrapping

- The essence of the pilot is bootstrapping DIRAC on a machine
- In all cases, we run a short script which either is the pilot or fetches the pilot
  - Constrained by the VM case where all you start with is a single user-data file
- Pilot 3.0 does this by

```
wget --recursive https://some.wh.ere/some/directory/  
./dirac-pilot.py ...
```

- In SiteDirector cases this will be included in the job script
- In VM cases this is included in the user-data file

# Bootstrapping -> Universality

- This procedure means the same pilots are used for both SiteDirector and VM cases
- But also that the same version is used in both cases, and across all of the pilots starting at the same time
  - It's even later Late Binding
  - Avoids the problem with old versions of the pilot code waiting in input sandboxes on grid CEs
- It also makes it very easy for XyzDIRAC instances to add more pilot commands
  - Just drop them into the https directory
  - Or anything else that XyzDIRAC instance needs

# Bootstrapping configuration

- For the Vacuum VM case, still have a configuration bootstrapping issue
  - Generic pilot code needs to know the CS URL etc of this XyzDIRAC instance, DIRAC version, ...
  - Needs to know the site it's running at (eg may need to run different commands at different types of site)
- Pilot 3.0 includes a file pilot.json in the directory wget fetches
- This is minimal dump of the CS with enough info to configure the pilot
  - DIRAC version, commands to run, Setup, and all the CE to Site mappings
- Try to avoid need to hardcode things in XyzDIRAC
  - Name=Value pairs can also be passed to pilot commands

# Pilot logging

- Currently pilot logs are retrieved by SiteDirectors using the mechanism for that type of CE (eg CREAM API)
  - Not uniform. Not always implemented.
  - For VMs, an ad-hoc log upload is used (Vac Depository)
- Pilot 3.0 introduces a message-based logging system (cf syslog)
  - Suitable for storing centrally in ElasticSearch
  - Can also be used for storing whole log files
- The same system across all ways of starting the pilot
- Can be presented in a uniform way to people on shift and users in the XyzDIRAC Portal instance
- Can be used before the pilot starts (eg by the SiteDirector) so pilot submissions that fail to run are visible

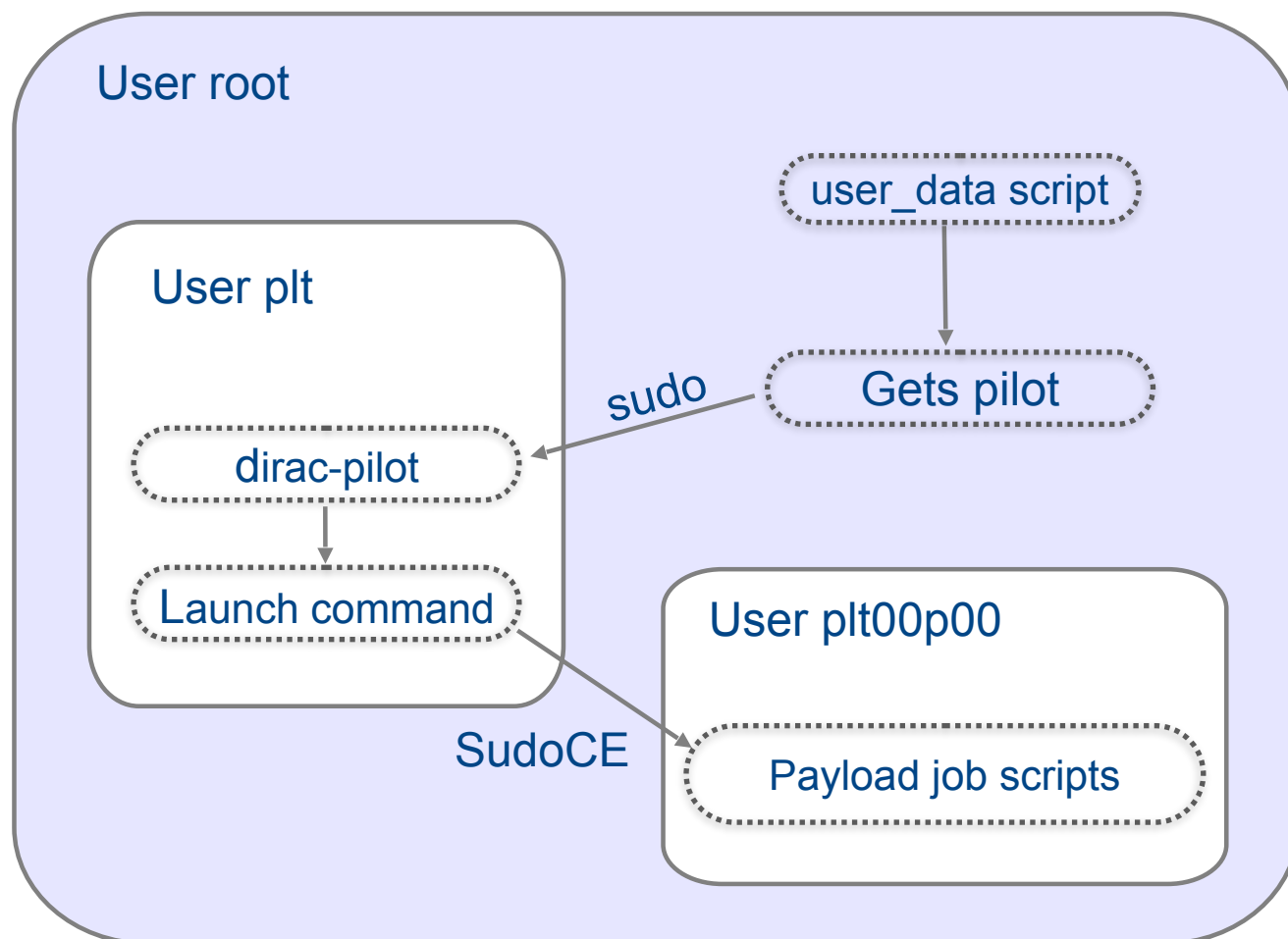


# Pilot 3.0 repository

- Decision was made early on to decouple Pilot 3.0 from main DIRAC repo and versioning
- Pilot has different constraints:
  - Has to run everywhere, before anything is installed
  - Probably has to use native Python
  - DIRAC version is indicated late, by configuration, perhaps on a per-site or CE basis for development?
- So use DIRACgrid/Pilot repo in GitHub
- Aim for very simple structure:
  - Tags with versions; production and development branches?

# Pilot 3.0 VMs isolation

- Uses unix accounts and sudo to isolate root vs pilot vs payloads
- SudoComputingElement does the sudo for the payloads
- Requires account creation per payload
  - Easy within VMs but doesn't generalise to batch cases





# Job isolation and Singularity

- WLCG has working group on Traceability and Isolation
- Has settled on Singularity lightweight container framework
  - Which has a lot of traction in HPC and growing support from CMS and ATLAS
- Singularity does not require a daemon (cf Docker)
- In sufficiently late Linux kernels, you do not need any setuid wrapper
  - Not yet in RHEL 7.x
- By adding support in DIRAC (SingularityComputingElement?) we can have a single framework for pilots in VMs and in batch
  - This pilot structure will be very similar to VMs

## Other containers

- Some work has been done to run Pilot 3.0 for LHCb in Docker containers
  - Yandex SkyGrid and Mesos-managed farm at RAL
  - Slightly different approaches wrt cvmfs
- We've started producing a general Pilot 3.0 Docker container definition based on CernVM root filesystem (ie via cvmfs)
  - Will be an instance of a more generic Vacuum Container, which Vac will be able to run
- All of the above use the VM user\_data file as a starting point
  - Lots of commonality



# Deployment

- All the LHCb VMs use Pilot 3.0
  - 6 Vac sites, 5 Vcycle sites; BOINC
  - The two Docker-based LHCb sites (Yandex Skygrid and RAL) use Pilot 3.0
- Pilot 3.0 has been demonstrated running pilots and payloads in VMs for Belle II DIRAC and the GridPP DIRAC service
  - Expect the Pilot 3.0 implementation will become the default for GridPP DIRAC VMs this week
- Deployments are not truly universal yet
  - Some changes required here and there
  - However there don't seem to be any requirements that will prevent convergence using Pilot 3.0 framework



# Summary

- Pilot 3.0 aims for universality
- Remove need to fork the pilot code wherever possible
- Make things configurable
- wget bootstrapping procedure will be common to Vacuum VMs and batch jobs from SiteDirectors
- General Pilot Logging improves uniformity and fills in gaps
- Pilot 3.0 builds on job/payload isolation of Pilot 2.0 VMs
  - Can be the basis for container based isolation in VMs and batch
- Work underway to extend to Docker containers as alternative to VMs
- Pilot 3.0 already in production for LHCb VMs. More coming ...